Teaching PROFESSOR new math

Holger Schulz, Juliane Müller, Sven Leyffer, Mohan Krishnamoorthy, Anthony Austin, Stephen Mrenna

MPI@LHC 2018 Perugia, 13 December 2018
Tuning

- Goal: best possible physics prediction of MC generator
- Realistic events contain physics at low scales where perturbation breaks down
- Rely on model assumptions that introduce many parameters
- Need to find “meaningful” settings
- Can be done manually but hard to do on a reasonable time-scale because of MC run-time and dimensionality of problem
Tuning with Professor in a nutshell

- Random sampling: $N$ parameter points in $n$-dimensional space
- Run generator and fill histograms (e.g. Rivet) trivial parallel
- Polynomial approximation per bin
- Construct goodness-of-fit measure

$$
\phi^2(\vec{p}) = \sum_b w^2_b \cdot \left( \frac{f_b(\vec{p}) - R_b}{\Delta^2} \right)^2
$$

- and numerically minimize with iminuit

In the following this will be called the “inner optimisation” problem
Inner and outer optimisation

- Incompatible datasets and mismodelling in MC generator necessitate introduction of tuning weights $w_b$
- Adjusting the weights has so far been a manual procedure
- The user would iteratively run the “inner optimisation“ and look at resulting plots
- We propose an automated procedure for this ”outer optimisation“:
  - Write goodness-of-fit in terms of histograms/observables
  - The parameter space is now the observable-weight space
  - Inner optimisation yields best fit point, $\hat{p}$, for given $
\\left\{w_\mathcal{O}\right\}$

$$\sum_{\mathcal{O}=1}^{N} w_\mathcal{O}^2 \cdot \sum_{b \in \mathcal{O}} \frac{(f_b(\hat{p}) - R_b)^2}{\Delta^2}$$

Minimisation yields $\hat{p} | \{w_\mathcal{O}\}$

- $\hat{p}$ is used to evaluate an objective function for the outer optimisation
Portfolio objective

- For given $\hat{p}$, we can calculate the per-observable goodness-of fit

$$\nu_\mathcal{O}(\hat{p}| \{w_\mathcal{O}\}) = \frac{1}{N_{\text{bins}}(\mathcal{O})} \sum_{b \in \mathcal{O}} \frac{(f_b(\hat{p}| \{w_\mathcal{O}\}) - R_b)^2}{\Delta^2}, \ \mathcal{O} = 1, \ldots, N$$

- With $N$ such measures, we can calculate mean and standard deviation (dropped argument $\hat{p}| \{w_\mathcal{O}\}$ for readability)

$$\mu = \frac{1}{N} \sum_{\mathcal{O}=1}^{N} \nu_\mathcal{O}$$

$$\sigma^2 = \frac{1}{N} \sum_{\mathcal{O}=1}^{N} [\nu_\mathcal{O} - \mu]^2$$

- And construct an objective function to minimise

$$\min_{w_\mathcal{O} \in [0,1]} \lambda \cdot \mu(\hat{p}| \{w_\mathcal{O}\}) + \sigma^2(\hat{p}| \{w_\mathcal{O}\}), \ \text{s.t.} \ \sum_{\mathcal{O}=1}^{N} w_\mathcal{O} = 1.$$
Outer optimisation

- Minimisation of portfolio objective is iterative, gradient free
- We train a radial basis function (RBF) and use it to walk through the weight space
- RBF minimisation alternates between local and global search
- Convergence is fast but depends on initial guess → multi-start approach (that’s ok since inner optimisation is really fast)
Evolution of inner optimisation

- This plot shows the $\hat{p}$ of the inner optimisation
- Shows the correlation of parameters
Comparison of results

Transverse $N$ density vs. $p_{T}^{\text{clus}}$, $\sqrt{s} = 900$ GeV

$\langle d^{2}N / d\eta d\phi \rangle$

$\langle d^{2}N / d\eta d\phi \rangle$

Transverse $\sum p_{T}$ density vs. $p_{T}^{\text{clus}}$, $\sqrt{s} = 900$ GeV

$\langle d^{2} \sum p_{T} / d\eta d\phi \rangle$

$\langle d^{2} \sum p_{T} / d\eta d\phi \rangle$

Transverse $\sum p_{T}$ density vs. $p_{T}^{\text{clus}}$, $\sqrt{s} = 7$ TeV

$\langle d^{2} \sum p_{T} / d\eta d\phi \rangle$

$\langle d^{2} \sum p_{T} / d\eta d\phi \rangle$

Transverse $N$ density vs. $p_{T}^{\text{clus}}$, $\sqrt{s} = 7$ TeV

$\langle d^{2}N / d\eta d\phi \rangle$

$\langle d^{2}N / d\eta d\phi \rangle$

MC/Data

Data

Portfolio

Standard

Random guess

$p_{T}$ (leading particle) [GeV]

$p_{T}$ (leading particle) [GeV]
Rational approximation

- Polynomial approximation does not capture $1/x$ behaviour well
- E.g. masses in propagators, MPI cut-off
- → Multivariate rational approximation $f(\vec{p}) = g(\vec{p})/h(\vec{p})$
- With $g, h$ being polynomials of order $m, n$
- N.b. polynomials are limit where $h(\vec{p})$ is constant ($n = 0$)
Spurious poles and selecting m and n

- Especially in presence of noisy data: spurious poles
- For numerical reasons, denominator polynomial can have roots
- Root finding greatly helped by knowledge of gradient
- We have a brute force and a smart method to deal with that:
  - Compute all possible approximations \((m, n)\), pick the best without pole
  - Instead of linear algebra, use non-linear optimisation with constraints to solve \(\|f(\vec{p})h(\vec{p}) - g(\vec{p})\|\), iteratively adding constraints on poles when found
Tuning uncertainties (with A. Buckley)

So far: exploit minimiser covariance matrix in parameter space

- Linear algebra to find principal directions
- In each direction: find intercept with $\chi^2$ contour
- Problem: $\Delta \chi^2 = 1$ recipe does not always work
- $\rightarrow$ Ad-hoc definition of $\Delta \chi^2$ by looking at plots (e.g. A14)
Bootstrapping the goodness-of-fit

To overcome the ad-hoc nature, obtain the actual distribution of the goodness-of-fit measure, $\Phi^2$

- Tuning replicas — smear data within its uncertainties, run minimisation for each replica and record $\Phi^2$

- Problem: coverage test revealed that there is a 25% chance the central tuning is outside the 68th percentile of $\Phi^2$
Latest attempt

- Still do the tuning replicas
- Forget about $\Phi^2$, $\chi^2$ values, forget about minimiser covariance
- Instead:
  - Calculate covariance in parameter space from tuning replicas
  - Linear algebra to find principal directions, eigenvalues for aspect ratio of ellipsoid
  - Find ellipsoid that contains 68.8\% of tuning replicas
  - Intercept with principal axes gives “Eigentunes”
Construction

# Bin

Entries
Truth
Central tune ($\chi^2 = 17.9$)
'Data'

14/15
Construction

![Graph showing data points and trends](image)

- **Truth**
- **ET1+**
- **ET1-**
- **ET2+**
- **ET2-**

Central tune ($\chi^2 = 17.9$)

'Data'
Summary

▶ Algorithm for a more automated tuning:
  - Outer optimisation loop in the weight space
  - Minimisation of portfolio objective function

▶ Algorithm for multivariate rational approximation
  - Rational approximation are a natural extension of polynomials used so far
  - Better quality interpolations
  - Spurious poles are a bit of a nuisance but can now be dealt with

▶ New suggestion to get tuning uncertainties
  - Tuning replicas through data smearing
  - Confidence ellipsoid construction and intercept with principal directions